*ATECC508A*

# INTRODUÇÃO

Este documento explica como utilizar o software Cryptotronix EClet para configurar o chip Atmel ATECC508A em uma raspberry pi.

# ARQUIVOS NECESSÁRIOS

Será necessário o download do driver:

<https://github.com/cryptotronix/EClet>

pi@raspberrypi:~ $ git clone https://github.com/cryptotronix/EClet.git

INSTALAÇÃO

Instale build-essential, autotools-dev, automake, autoconf, libtool, libxml2-dev, check, texinfo, and libgcrypt (libgcrypt11-dev on Debian variants)

pi@raspberrypi:~ $ sudo apt-get install build-essential autotools-dev automake autoconf libtool libxml2-dev check texinfo libgcrypt11-dev

Após isso rode o arquivo **./autogen.sh .** Isso irá gerar o arquivo README necessário a partir do arquivo README.md assim como a versão beta requerida da biblioteca baixada anteriormente.

pi@raspberrypi:~ $ sudo ./autogen.sh

pi@raspberrypi:~ $ sudo make install

# CONFIGURAÇÔES INICIAIS

Inicialmente é necessário executar **/dev/i2c\*** para o acesso. É possível mudar isto adicionando seu usuário para o grupo I2C utilizando: **sudo usermod -aG i2c user** ou **sudo chmod o+rw /dev/i2c\* .**

Existem alguns comandos suportados:

**eclet state** : verifica se está em estado de fábrica.

pi@raspberrypi:~ $ eclet state

Factory

**eclet personalize** : em caso de sucesso este comando não retorna nada, sua função é configurar todos os slots (0-16) para a chave privada. O slot 8 é reservado para uso futuro.

pi@raspberrypi:~ $ eclet personalize

pi@raspberrypi:~ $ eclet state

Personalized

**eclet random** : Até que você tenha personalizado o seu dispositivo, o gerador de número aleatório irá retornar um número com padrões contendo FF e 00.

pi@raspberrypi:~ $ eclet random

7E74FBFE17A10026124B092C9727416D6C28B38FA5D598E0711A9FCDFCDEC408

**eclet serial-num** : Retorna o serial do dispositivo.

pi@raspberrypi:~ $ eclet serial-num

0123XXXXXXXXXXXXEE

**eclet gen-key** : Irá criar uma chave privada com 256 bits e irá retornar uma chave pública de formato 0x04 + X +Y. Caso o comando seja dado mais vezes a chave pública irá ser alterada.

pi@raspberrypi:~ $ eclet gen-key

042A02563C69C4CB5356BD8BBA96B1559D5FE799B6C497B128F42F8CB31FF751BF40361F0970DF48979B3E18A1039047299A1C3D7B809EAA2F844F0D75C479BDB9

**eclet sign -f ChangeLog** : Cria uma assinatura ECDSA. O arquivo pode ser especificado com -f.

pi@raspberrypi:~/Eclet $ eclet sign -f ChangeLog

5D1BD52DD2294CA1518E2520CFBF518AD6DD48AA57978BCAF91919E51E5FE3CA2B1EFC8427AC7EC499DF340B197319545FCB2A7F87D981F6712365CF3E6872B3

**eclet verify** : Verifica a assinatura ECDSA usada no dispositivo. Se estiver correta, nada acontecerá, caso contrário uma mensagem de falha na verificação acontecerá.

pi@raspberrypi:~/Eclet $ eclet verify -f ChangeLog -–signature 5D1BD52DD2294CA1518E2520CFBF518AD6DD48AA57978BCAF9191

9E51E5FE3CA2B1EFC8427AC7EC499DF340B197319545FCB2A7F87D981F6712365CF3E6872B3 -–public-key 04DD24F3770BD11A6A465F37D1D6CEEA58F8B8E

1B85A5D3D665A0382BCBCDDC81321458F8A4FE5777BA0D508780A2476A23434B89BE2BBD5B8CF574348A15F1982

**eclet offline-verify-sign** : Verifica a assinatura ECDSA usada, sem a necessidade do dispositivo. Se estiver correta, nada acontecerá, caso contrário uma mensagem de falha na verificação acontecerá.

pi@raspberrypi:~/Eclet $ eclet offline-verify-sign -f ChangeLog -–signature 5D1BD52DD2294CA1518E2520CFBF518AD6DD48AA

57978BCAF91919E51E5FE3CA2B1EFC8427AC7EC499DF340B197319545FCB2A7F87D981F6712365CF3E6872B3 -–public-key 04DD24F3770BD11A6A465F37D1

D6CEEA58F8B8E1B85A5D3D665A0382BCBCDDC81321458F8A4FE5777BA0D508780A2476A23434B89BE2BBD5B8CF574348A15F1982

# OPÇÕES

As opções podem ser visualizadas utilizando o comando **–-help .**